

Instituto Politécnico do Porto

**Documentos Inteligentes**

**António José Rocha de Oliveira**

Trabalho apresentado para a satisfação parcial dos requisitos do grau de  
Especialista

Departamento de Engenharia Informática do Instituto Superior de  
Engenharia do Porto

Porto – Outubro de 2010

# 1 Resumo

Ao longo deste trabalho será apresentada uma solução para sistemas de informação empresarial no domínio da Qualidade. Desenvolvida num ambiente real com recurso a plataforma .NET, a sua eficácia foi posta à prova em várias implementações.

Várias organizações necessitam de implementar processos baseados num documento único, que deve conter o registo de toda a informação, mas que exige a intervenção de diferentes colaboradores. Esta necessidade varia muito entre organizações.

Para dar resposta a este problema foi desenvolvido um sistema, denominado de Documentos Inteligentes, que, através de várias ferramentas, permite configurar processos, sem recorrer a codificação, baseados no registo de informação, workflow, controlo de acessos e alertas.

É justamente a descrição dos vários componentes tecnológicos utilizados para dar resposta aos problemas, que se irá efectuar neste trabalho.

Em termos de desenvolvimento de software foram abordados conceitos práticos de engenharia de domínio, RAD, arquitecturas por camadas, abstracção e workflow que possibilitaram uma maior flexibilidade do sistema desenvolvido e que conduz, simultaneamente, a uma rápida implementação de soluções de software baseadas nesta área.

Palavras Chave: .NET; Engenharia de domínio, Qualidade, arquitecturas de software por camadas; Metadados, sistemas de informação empresariais; Configuração;

### 3 Índice geral

<b>1</b>	<b>Resumo .....</b>	<b>ii</b>
<b>3</b>	<b>Índice geral .....</b>	<b>iii</b>
<b>4</b>	<b>Lista de figuras .....</b>	<b>v</b>
<b>5</b>	<b>Lista de tabelas .....</b>	<b>vi</b>
<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	Contexto .....	1
1.2	O problema por exemplo .....	1
1.3	A solução .....	3
1.4	Estrutura do trabalho .....	6
<b>2</b>	<b>B-Simple Framework .....</b>	<b>7</b>
<b>3</b>	<b>B-Designer .....</b>	<b>10</b>
3.1	O desenho .....	10
3.2	Vista da base de dados .....	12
3.3	Propriedades .....	12
3.4	Componentes visuais .....	13
3.4.1	Anexar Documentos .....	13
3.4.2	Data .....	13
3.4.3	Anexar Documento .....	13
3.4.4	Campo de aprovação .....	14
3.4.5	Seleção exclusiva .....	14
3.4.6	Campo externo .....	14
3.4.7	Campo Numérico .....	14
3.4.8	Campo Sim/Não .....	14
3.4.9	Campo Texto .....	14
3.4.10	Grupo .....	15
3.4.11	Processo Seguinte .....	15
3.4.12	Tabulador .....	15
3.4.13	Lista de Seleção .....	16
3.4.14	Texto .....	16
3.4.15	Editor HTML .....	17
3.5	Outras opções .....	17
3.6	Ambiente Windows e Web .....	18
<b>4</b>	<b>B-Config .....</b>	<b>20</b>
4.1	Workflow .....	20
4.2	Processos seguintes .....	22
4.3	Possibilidades .....	22
4.4	Notificações .....	23
4.5	Avançado .....	24
<b>5</b>	<b>CLASSES E COMPONENTES .....</b>	<b>25</b>
5.1	Classes .....	25
5.1.1	Classe Principal .....	26
5.1.2	Classe FormDin .....	26
5.1.3	Classe FormDinDesign .....	26
5.1.4	Classe FormDinRules .....	27
5.1.5	Classe DBHelper .....	28
5.1.6	Classe External e Extensibilidade .....	28

---

5.2 Diagrama de Sequência .....	28
5.3 Diagrama de Componentes .....	30
<b>6 MODELO DE DADOS.....</b>	<b>32</b>
6.1 Tabelas de configuração .....	32
6.1.1 Tabela ModConfig .....	33
6.1.2 Tabela ModLink.....	35
6.1.3 Tabela ModConfigWorkFlow .....	35
6.1.4 Tabela ModCamposAprov .....	36
6.1.5 Tabela ModCamposAprovNotific .....	38
6.1.6 Tabela ModCamposAprovPoss .....	39
6.1.7 Tabela ModCamposAprovPerm.....	39
6.2 Tabelas por documento inteligente .....	41
6.2.1 Tabela <Modulo> .....	41
6.2.2 Tabela <Modulo>MultiAprov.....	43
6.2.3 Tabela <Modulo>RegUt .....	43
<b>7 CONCLUSÕES.....</b>	<b>45</b>
7.1 Ferramentas e metadados .....	45
7.2 Mercado .....	45
7.3 Considerações finais .....	46

## 4 Lista de figuras

Figura 1.1 – Exemplo de ocorrência .....	2
Figura 1.2 – Esquema geral da solução implementada.....	3
Figura 1.3 – Exemplo do documento ocorrência .....	4
Figura 2.1 – B-Simple Framework.....	7
Figura 3.1 – B-Designer.....	10
Figura 3.2 – Exemplo de áreas de aprovação.....	11
Figura 3.3 – Exemplo de propriedades .....	12
Figura 3.4 – Anexar documentos .....	13
Figura 3.5 – Anexar documento .....	13
Figura 3.6 – Campo de aprovação.....	14
Figura 3.7 – Seleção exclusiva.....	14
Figura 3.8 – Campo externo .....	14
Figura 3.9 – Grupo.....	15
Figura 3.10 – Processo seguinte.....	15
Figura 3.11 – Tabulador.....	16
Figura 3.12 – Lista de detalhe.....	16
Figura 3.13 – Editor HTML.....	17
Figura 3.14 – Opções avançadas .....	17
Figura 3.15 – Ocorrência em ambiente Windows.....	18
Figura 3.16 – Ocorrência em ambiente web.....	19
Figura 4.1 – Configuração do workflow .....	20
Figura 4.2 – Configuração dos processos seguintes.....	22
Figura 4.3 – Configuração das possibilidades por actividade .....	22
Figura 4.4 – Configuração das notificações .....	23
Figura 4.5 – Configuração das opções avançadas.....	24
Figura 5.1 – Diagrama das classes principais .....	25
Figura 5.2 – Diagrama de sequência de criação de um registo.....	29
Figura 5.3 – Diagrama de sequência de criação de um registo.....	30
Figura 6.1 – Diagrama das classes persistentes de configuração.....	32
Figura 6.2 – Diagrama das classes persistentes modelo .....	41

## 5 Lista de tabelas

Tabela 5.1 – Principais métodos do FormDin.....	26
Tabela 5.2 – Principais métodos do FormDinDesign.....	27
Tabela 5.3 – Principais métodos do FormDinRules.....	27
Tabela 5.4 – Componentes envolvidos .....	30
Tabela 6.1 - Campos do ModConfig .....	33
Tabela 6.2 - Campos do ModLink .....	35
Tabela 6.3 - Campos do ModConfigWorkFlow.....	35
Tabela 6.4 - Campos do ModCamposAprov .....	36
Tabela 6.5 - Campos do ModCamposAprovNotific.....	38
Tabela 6.6 - Campos do ModCamposAprovPoss .....	39
Tabela 6.7 - Campos do ModCamposAprovPerm .....	40
Tabela 6.8 - Campos do <Modulo> .....	42
Tabela 6.9 - Campos do <Modulo>MultiAprov .....	43
Tabela 6.10 - Campos do <Modulo>RegUt.....	43

# 1 INTRODUÇÃO

## 1.1 Contexto

Diversas organizações certificadas ou com processo de certificação em curso, têm nos seus processos de negócio uma enorme diversidade de formulários para efectuar os registos de dados. Estes registos têm, normalmente, um conjunto de regras pré-estabelecidas sobre quem intervém, onde intervêm e quando intervêm.

Constata-se que estes formulários variam de empresa para empresa. Por exemplo, o registo de uma ocorrência na empresa A pode ser muito diferente da empresa B.

Por outro lado, as empresas, na sua maioria, não estão dispostas a pagar o desenvolvimento de um software específico para resolverem este tipo de problemas.

É neste contexto que surgiu o software B-Quality. Este foi criado com um nível elevado de abstracção por forma a ser possível adaptá-lo a qualquer tipo de organização. Neste documento será explicado, não todo o software, mas a área que mais concretamente dá resposta ao problema descrito acima: os documentos inteligentes.

## 1.2 O problema por exemplo

Para mostrar alguns dos problemas envolvidos, bem como a solução encontrada, será usado um documento exemplificativo utilizado por uma organização (fig. 1.1).

Neste documento são registadas as ocorrências da organização. Podem-se identificar três áreas/actividades:

- Abertura da ocorrência
- Classificação de ocorrência
- Fecho da ocorrência

Cada actividade pode ter diferentes pessoas envolvidas. Por exemplo, a actividade 2 é realizada pelo responsável do sector onde aconteceu a ocorrência, enquanto a actividade 3 é concluída pelo director da qualidade.

É também possível verificar um workflow sequencial, que inicia na Abertura, passa para a Classificação e termina no Fecho.

As diferentes possibilidade de workflow, bem como os dados a registar em cada actividade varia por empresa, especialmente entre empresas de diferentes ramos de negócio.

Empresa Fictícia SA

**Ocorrência**

Sector \_\_\_\_\_ Número \_\_\_\_\_

Descrição

Análise das causas

Assinatura \_\_\_\_\_ Data \_\_\_\_/\_\_\_\_/\_\_\_\_

Encerramento pelo Responsável de sector

Descrição \_\_\_\_\_

Gravidade	<input type="radio"/> Ligeira	Tipo	<input type="radio"/> Operacional
	<input type="radio"/> Grave		<input type="radio"/> Técnica
	<input type="radio"/> Muito grave		<input type="radio"/> Equipamento _____
			<input type="radio"/> Aprovisionamento
			<input type="radio"/> Manutenção
			<input type="radio"/> Outra

Assinatura \_\_\_\_\_ Data \_\_\_\_/\_\_\_\_/\_\_\_\_

Verificação pelo Dir. da Qualidade

Assinatura \_\_\_\_\_ Data \_\_\_\_/\_\_\_\_/\_\_\_\_ ACP \_\_\_\_\_

Anexos \_\_\_\_\_

RRR-F044

Pág. 1 de 1

Figura 1.1 – Exemplo de ocorrência

Esta diversidade de situações tornava praticamente impossível criar um módulo que resolvesse os problemas de todas as organizações. A solução passaria por encontrar



um software que, em cada implementação, possibilitasse a criação de cada um destes documentos à medida das necessidades.

Tendo a *software house* como objectivo a distribuição do produto por outras entidades empresariais, era imperativo criar uma solução que permitisse independência a esses implementadores nas suas implementações.

### 1.3 A solução

Para resolver este desafio foi idealizado um sistema que permite, em cada implementação, configurar todo o processo através de um conjunto de ferramentas (ver figura seguinte) assente na seguinte estrutura:

- Desenhador de interfaces de utilizador (B-Designer)
- Configurador de processos (B-Config)
- Gestor de processos que interpreta os metadados gerados pela B-Designer e pelo B-Config e cria os módulos em run-time.
- Gestor de alertas encarregue de enviar os alertas aos destinatários.



Figura 1.2 – Esquema geral da solução implementada

O B-Designer é um desenhador de ecrãs onde é possível criar uma interface de utilizador semelhante ao documento que necessita de ser informatizado. Permite colocar e posicionar os mais diversos componentes como caixas de texto, datas,

grelhas, entre outros. Permite ainda estabelecer a ligação entre os dados introduzidos e a sua correspondência na base de dados. Tem como resultado um conjunto de metadados que são guardados na base de dados.

O B-Config, ou configurador de processos, permite configurar o workflow existente no módulo, os intervenientes, as notificações e a forma como o módulo irá assentar na framework da B-Simple (framework de desenvolvimento de software que será descrita mais à frente no capítulo 2). Estas definições são guardadas na base de dados.

O Gestor de Processos, sempre que invocado, apresenta ao utilizador um módulo para introdução de informação de acordo com as configurações feitas pelo B-Designer e pelo B-Config. O módulo é ainda apresentado consoante o estado do workflow e o perfil do utilizador. O utilizador pode concluir uma actividade através de um botão disponibilizado para o efeito (pontos de aprovação).

Na figura abaixo podemos ver o resultado apresentado ao utilizador resultante da configuração do documento da ocorrência.

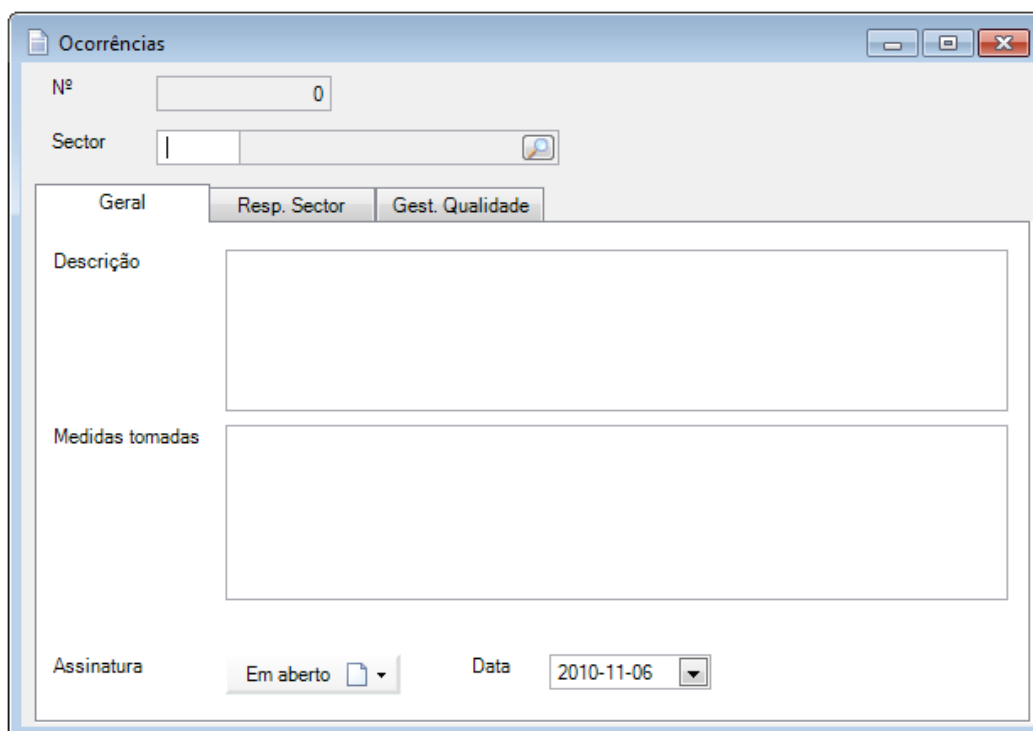


Figura 1.3 – Exemplo do documento ocorrência

O Gestor de Alertas monitoriza os processos e alerta os colaboradores sobre o seu estado, tendo em conta as definições feitas nos alertas. Tem como objectivo alertar os colaboradores para intervir nas actividades envolvidas e para os atrasos das mesmas.

Todos os elementos desenvolvidos combinam entre si por forma a construir um processo de negócio completo.

Ao longo dos próximos capítulos será detalhado cada um destes componentes, bem como os restantes envolvidos no processo.

## **1.4 Participação**

Interessa aqui referir o nível de participação do autor nesta solução nas várias fases do projecto. As percentagens apresentadas, são valores aproximados:

- Visão: 100%
- Arquitectura: 100%
- Análise: 90%
- Programação: 5%
- Testes: 30%
- Director de Projecto: 100%

## 1.5 Estrutura do trabalho

Este trabalho é apresentado em sete capítulos, estando a estrutura essencialmente condicionada pela organização do software desenvolvido.

No primeiro capítulo é feita a introdução ao problema abordado, bem como uma breve descrição da solução encontrada.

No segundo capítulo é apresentada a Framework B-Simple, onde assentam os componentes desenvolvidos.

No terceiro capítulo é apresentado o B-Designer, nomeadamente a sua arquitectura e funcionalidades.

No quarto capítulo é apresentado o B-Config, nomeadamente a forma como é criado um workflow e as várias possibilidades de configuração de um processo.

No quinto capítulo são descritas as principais classes e componentes que suportam a estrutura do trabalho.

No sexto capítulo, é detalhado o modelo de dados de suporte a todos componentes desenvolvidos no trabalho.

No sétimo e último capítulo é feita a conclusão do trabalho frisando os objectivos, a solução e a diversidade de implementações atingida.

## 2 B-Simple Framework

A B-Simple Framework é uma framework de desenvolvimento de software onde assentam todas as aplicações da B-Simple.

Interessa aqui referir de forma resumida os princípios de funcionamento da B-Simple Framework, para que se perceba como encaixa toda a arquitectura de software da B-Simple e mais concretamente o projecto apresentado neste trabalho.

A B-Simple Framework define um conjunto de regras arquitecturais às quais todos os módulos devem obedecer e disponibiliza um conjunto de componentes e processos que são transversais a todos os módulos de uma solução. Os seus principais pilares podem ser observados na figura abaixo.

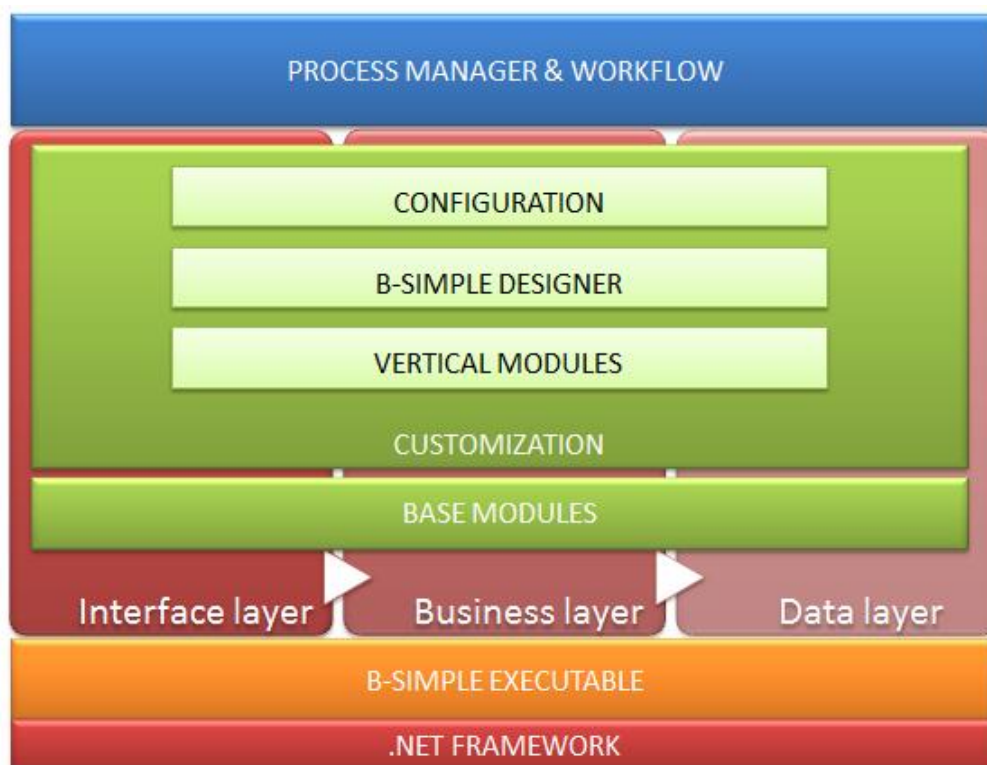


Figura 2.1 – B-Simple Framework

A arquitectura é constituída por um executável (.exe) que permite orquestrar todos os componentes de uma solução. Este executável, bem como todo o desenvolvimento, assenta sobre a .Net framework da Microsoft.

A arquitectura tem como requisito que todos os módulos sejam desenvolvidos em três camadas, tendo como vantagem a separação de conceitos:

- Interface layer – onde residem os serviços de apresentação e interacção com o utilizador. Todo o processamento que tem a ver com os ecrãs do utilizador é feito aqui;
- Business layer – Efectua o tratamento das regras do sistema;
- Data Layer - Estabelece a ponte entre as regras de negócio e a base de dados. Gere todos os pedidos de informação à base de dados (inserir, alterar, eliminar e consultar informação).

Neste tipo de arquitectura, as camadas comunicam entre si através de uma ordem que não pode ser alterada: a camada de apresentação invoca funcionalidades da camada de negócio, que por sua vez poderá ter necessidade de chamar funcionalidades da camada de dados. Não é possível a camada de apresentação invocar directamente a camada de dados ou as regras de negócio invocarem objectos da apresentação.

Qualquer módulo que assente sobre esta Framework é obrigado a herdar classes base disponibilizadas pela arquitectura que forçam automaticamente o desenvolvimento desses módulos por camadas.

A herança destes componentes disponibiliza, de uma forma simples, um conjunto de boas práticas que aceleram o desenvolvimento, garantem a qualidade das soluções e a uniformidade do desenvolvimento, tais como:

- Segurança de acessos
- Modos de funcionamento (criar, alterar, anular, consultar e imprimir)
- Componentes visuais
- Acessos a dados
- Logging
- Etc.

A Framework contém um conjunto de conceitos que permitem criar ou adaptar uma solução a uma determinada organização. Estes conceitos de extensibilidade assentam essencialmente sobre:

- Configuration – A framework permite configurar uma grande parte de conceitos disponibilizando ao implementador módulos de parametrização para o efeito;
- B-Simple Designer – Criação rápida de processos de negócio (a ver em detalhe mais à frente neste trabalho);
- Vertical modules – Permite o acoplamento de módulos de raiz, desde que estes obedeçam às regras da framework. Os módulos são posteriormente configurados na framework e instanciados dinamicamente por *reflection*.

Por forma a relacionar todos os módulos e garantir que os processos fluam de acordo com as necessidades de uma organização, foi criado um configurador de processos. Este configurador de processos será detalhado mais à frente no âmbito dos documentos inteligentes.

### 3 B-Designer

O B-Designer é um módulo que permite desenhar o aspecto final do documento inteligente (ecrã final do utilizador) e configurar a lógica de ligação às restantes camadas da arquitectura.

O módulo é constituído essencialmente por quatro áreas, conforme é possível observar na figura seguinte:

1. O desenho do documento inteligente
2. Vista da base de dados
3. Propriedades do documento inteligente
4. Barra de ferramentas

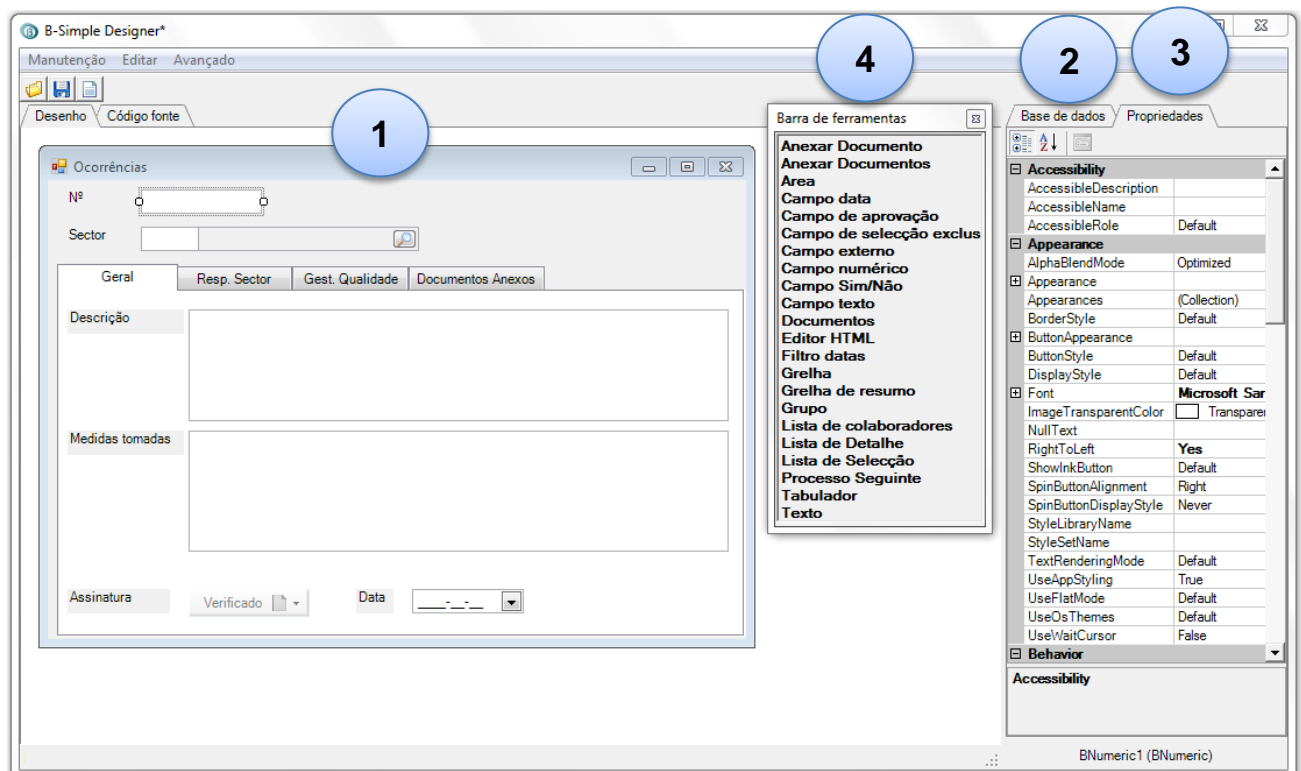


Figura 3.1 – B-Designer

#### 3.1 O desenho

Na área de desenho é dada a liberdade ao utilizador para dispor no ecrã os componentes necessários no módulo. Para colocar os componentes no formulário basta fazer um duplo click no componente que se pretende na barra de ferramentas.



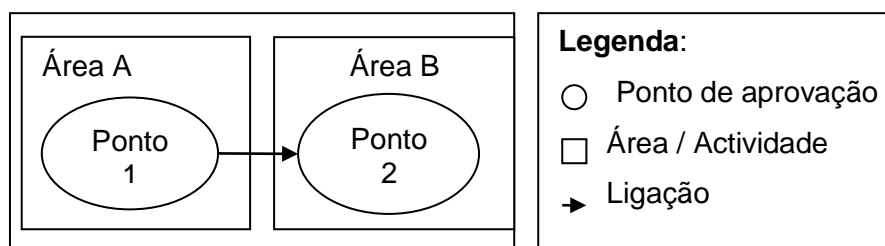
Para cada componente, existente no formulário, é dada a possibilidade de ajustar a sua posição com um simples drag & drop, ajustar a altura, ajustar o comprimento, bem como muitas outras propriedades (a lista de propriedades apresentada é função do componente seleccionado).

O formulário pode suportar várias actividades de um workflow. É aqui que é dada a indicação de quais os campos que pertencem às diferentes actividades para desta forma a aplicação ter a indicação de quais inputs aceitar em cada actividade. Para isso é necessário definir as áreas (ou actividades) e os pontos de aprovação existentes, tendo em conta que todos os campos que pertencem à mesma área dependem do ponto de aprovação que se encontra dentro da mesma, para estarem activos ou não.

É necessário ter em atenção que:

- Cada área só pode conter um campo de aprovação;
- As áreas ficam activas de forma sequencial de acordo com as ligações entre os pontos de aprovação especificadas no workflow.

Para melhor compreensão veja-se o seguinte exemplo:



Passando o exemplo para o documento inteligente poderíamos ter algo do género:

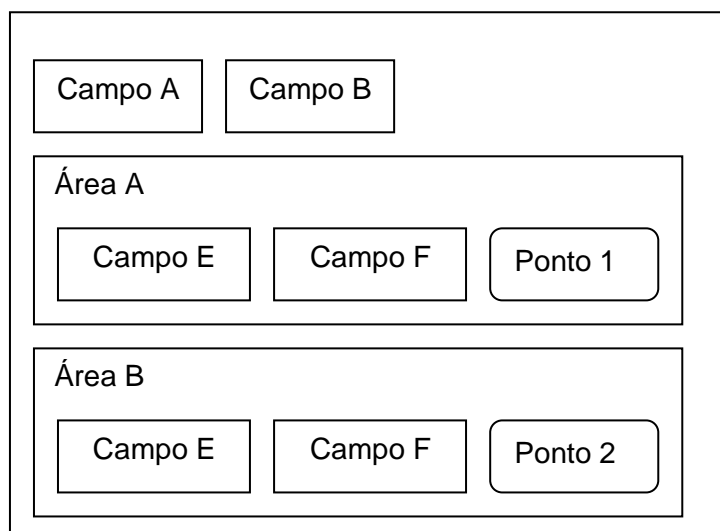


Figura 3.2 – Exemplo de áreas de aprovação

Neste exemplo, os campos A e B estão sempre activos, enquanto as áreas e todos os campos que se encontram dentro delas vão ficando activos à medida que os pontos de aprovação são resolvidos.

### 3.2 Vista da base de dados

Na vista da base de dados é possível ver todas as tabelas e respectivos campos, que constituem a base de dados ao qual está ligado o B-Designer.

Esta área serve também de facilitador para o desenho do formulário:

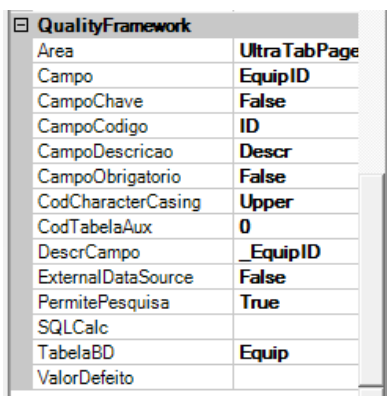
- Com um duplo click em cima de um campo é automaticamente colocado no ecrã um componente para input desse campo;
- Com um duplo click em cima da tabela, são automaticamente colocados no ecrã os componentes necessários para fazer o input de todos os campos da tabela.

Nesta área é também possível criar novas tabelas. Ao criar uma nova tabela via B-Designer, este cria automaticamente todos os campos obrigatórios que um documento inteligente deve ter.

### 3.3 Propriedades

Na área das propriedades é possível ajustar, entre outros, os atributos de um componente, como a fonte, o comportamento e a cor.

As propriedades mais importantes são as que têm influência no comportamento dos documentos inteligentes e que foram reescritas em todos os componentes aqui usados, nomeadamente o campo correspondente na base de dados. Estas propriedades variam de componente para componente. Abaixo segue um exemplo para o componente que tem um comportamento semelhante a uma dropdown.



Area	UltraTabPage
Campo	EquipID
CampoChave	False
CampoCodigo	ID
CampoDescricao	Descr
CampoObrigatorio	False
CodCharacterCasing	Upper
CodTabelaAux	0
DescrCampo	_EquipID
ExternalDataSource	False
PermitePesquisa	True
SQLCalc	
TabelaBD	Equip
ValorDefeito	

Figura 3.3 – Exemplo de propriedades

## 3.4 Componentes visuais

Para o B-Designer foi disponibilizada uma biblioteca de componentes que permitem, em conjugação entre si, criar um documento inteligente. De seguida são apresentados alguns desses componentes visuais.

### 3.4.1 Anexar Documentos

O componente Anexar Documentos, como o próprio nome indica, permite ao utilizador adicionar vários documentos, de qualquer tipo, a um documento inteligente, como por exemplo um PDF.

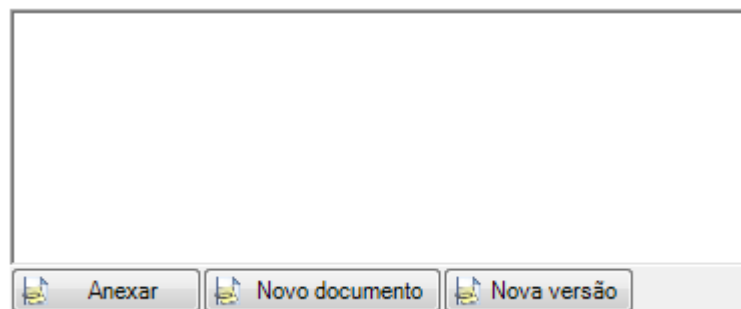


Figura 3.4 – Anexar documentos

### 3.4.2 Data

O componente Data permite fazer o input de uma data.

### 3.4.3 Anexar Documento

O componente Anexar Documento, como o próprio nome indica, permite ao utilizador adicionar um documento, de qualquer tipo, a um documento inteligente, como por exemplo um PDF.

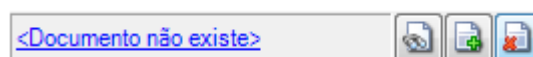


Figura 3.5 – Anexar documento

### 3.4.4 Campo de aprovação

O componente Campo de Aprovação permite aprovar/rejeitar uma actividade num workflow. É o botão usado em cada ponto de aprovação para condicionar o workflow.

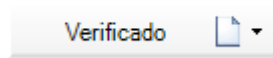


Figura 3.6 – Campo de aprovação

### 3.4.5 Selecção exclusiva

O componente Selecção exclusiva permite escolher uma opção entre várias apresentadas.

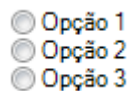


Figura 3.7 – Selecção exclusiva

### 3.4.6 Campo externo

O componente Campo Externo permite mostrar uma descrição associada a um campo que tem referência noutra tabela.

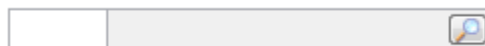


Figura 3.8 – Campo externo

### 3.4.7 Campo Numérico

O componente Campo Numérico permite configurar inputs do tipo numérico.

### 3.4.8 Campo Sim/Não

O componente Sim/Não permite efectuar inputs cujas opções sejam sim ou não.

### 3.4.9 Campo Texto

O componente Campo Texto permite configurar inputs do tipo texto.

### 3.4.10 Grupo

O componente Grupo permite definir uma área para agrupar vários componentes. O Grupo permite definir os componentes envolvidos por actividade.

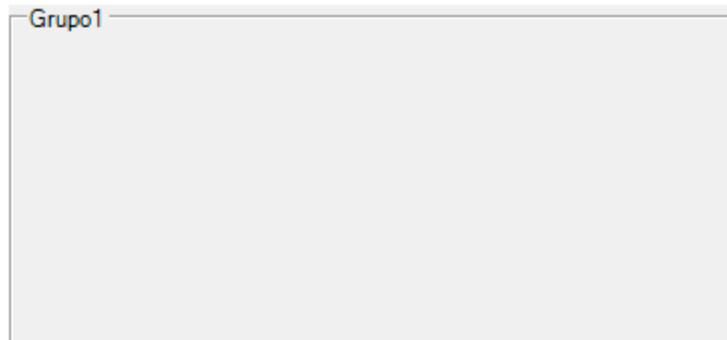


Figura 3.9 – Grupo

### 3.4.11 Processo Seguinte

O componente Processo Seguinte permite despoletar um outro processo na aprovação de um registo corrente. Exemplo: Ao registar uma ocorrência, no último passo poderá haver necessidade de gerar uma ACP (acção correctiva ou preventiva), que fique associada à ocorrência em questão. Assim no módulo de ocorrências existirá um botão de processo seguinte que, ao clicar, oferece ao utilizador a possibilidade de gerar a ACP. Caso a ACP seja gerada neste botão, ficará o número do novo registo e, ao clicar no botão, a aplicação abrirá automaticamente o registo em questão.

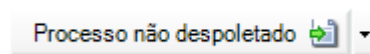


Figura 3.10 – Processo seguinte

### 3.4.12 Tabulador

Componente Tabulador permite agrupar os componentes em tabuladores. Para adicionar uma nova “página” de tabulador, é necessário seleccionar o tabulador e clicar nas propriedades em tabs como mostra a imagem seguinte.

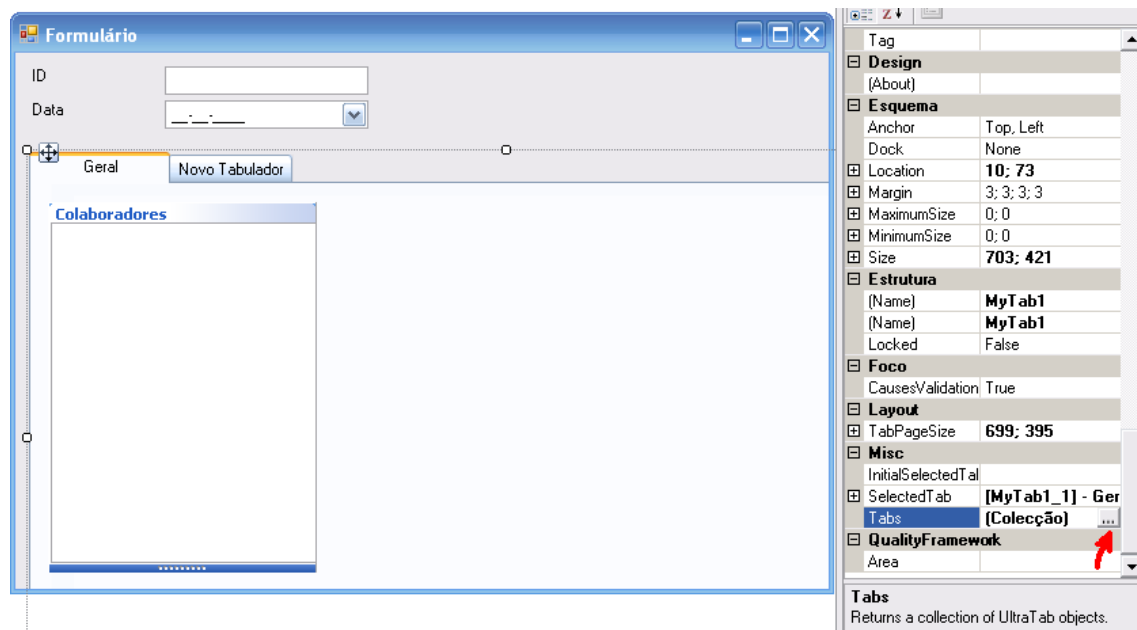


Figura 3.11 – Tabulador

### 3.4.13 Lista de Selecção

O campo Lista de Selecção permite adicionar várias entradas numa tabela de detalhe.

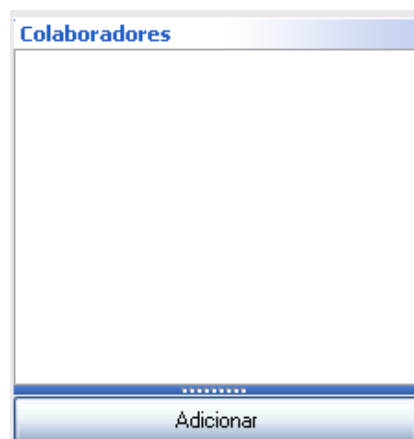


Figura 3.12 – Lista de detalhe

### 3.4.14 Texto

O componente Texto permite apresentar um texto (label).

### 3.4.15 Editor HTML

O Editor de HTML permite definir documentos de HTML onde é possível incluir texto, imagens, campos numéricos ou tabelas, de uma forma semelhante a um editor de texto.

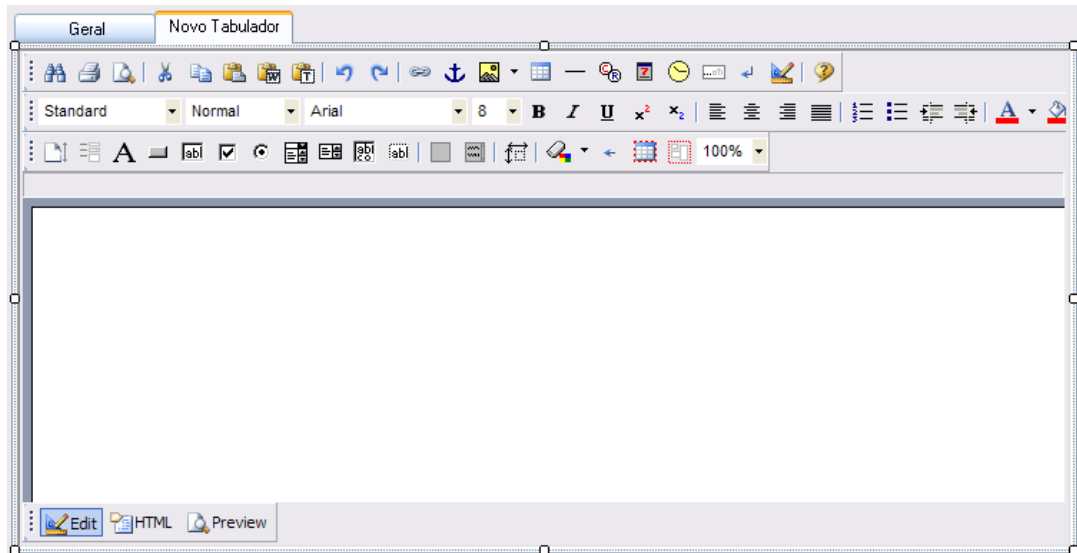


Figura 3.13 – Editor HTML

## 3.5 Outras opções

Além do referido atrás foram criadas outras opções que permitem facilitar o processo de criação de documentos inteligentes, tais como:

- Exportar: permite exportar para XML os metadados de um documento, bem como os scripts de criação das tabelas na base de dados;
- Importação: permite a criação de documentos inteligentes através dos scripts gerados na exportação;
- Base de dados: permite alternar entre base de dados.

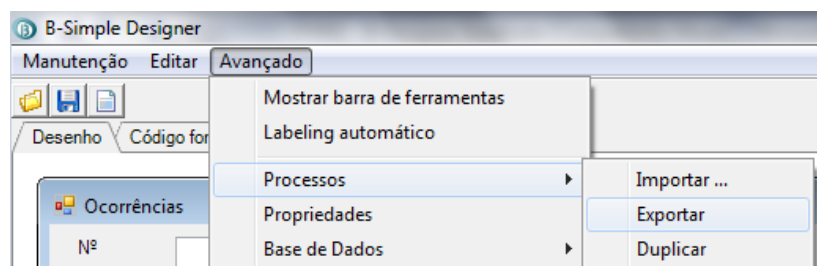


Figura 3.14 – Opções avançadas

### 3.6 Ambiente Windows e Web

As configurações permitem em tempo real a construção do documento e todas as suas regras em ambiente windows e web. Nas figuras seguintes é possível observar o resultado da configuração do documento Ocorrências nos dois ambientes.

The screenshot displays a web application window titled "B-Simple - Empresa Demo SA". The interface includes a menu bar with options: Ficheiro, Processos, Manutenções, Consultas, Inquéritos, Administração, and Ajuda. Below the menu is a toolbar with icons for Novo, Gravar, Cancelar, Alterar, Apagar, Anular, Pesquisar, Imprimir, and Ver antes. A navigation bar shows "1 de 1" with left and right arrows. The main content area features a form titled "Ocorrências". The form has input fields for "Nº" (containing "291") and "Sector" (containing "B" and "Produção"). Below these are three tabs: "Geral" (selected), "Resp. Sector", and "Gest. Qualidade". The "Geral" tab contains two text areas: "Descrição" (containing "Demo") and "Medidas tomadas" (containing "demo"). At the bottom of the form, there is an "Assinatura" section with a dropdown menu showing "Aprovado" and a "Data" field showing "2010-12-11". The footer of the application shows "Administrador" and two buttons: "Navegação" and "Ocorrências".

Figura 3.15 – Ocorrência em ambiente Windows



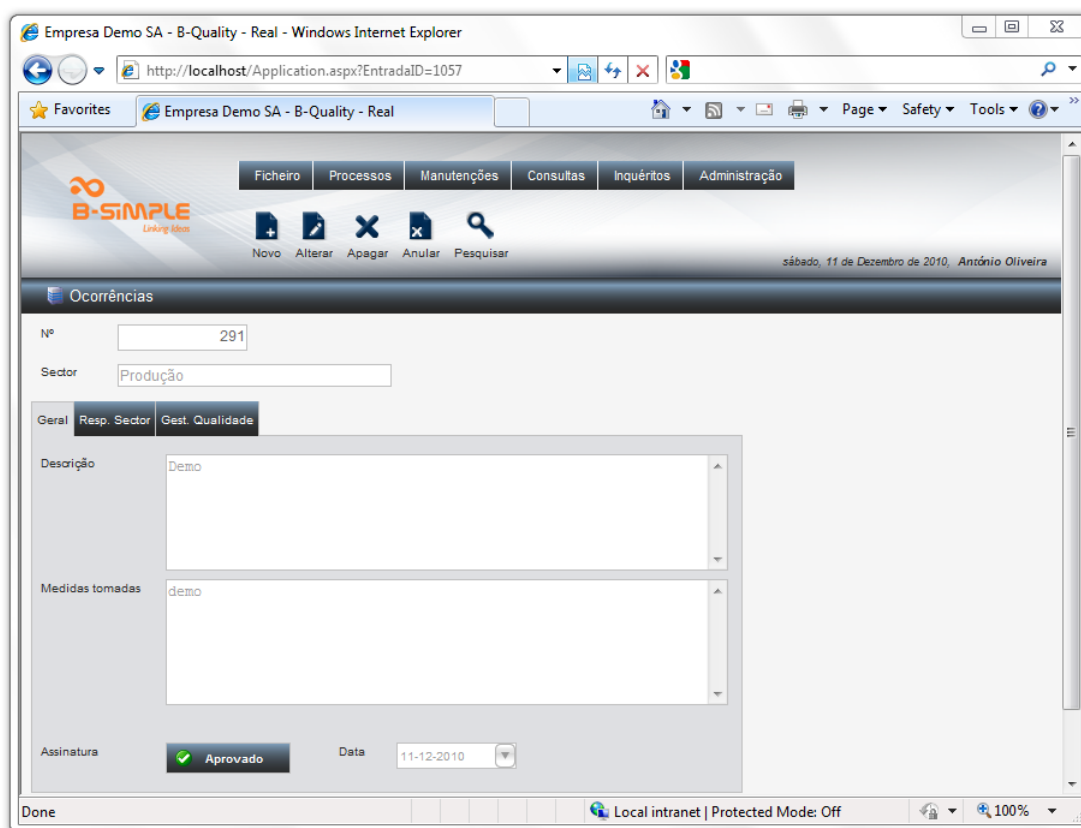


Figura 3.16 – Ocorrência em ambiente web

## 4 B-Config

O B-Config, ou gestor de processos, foi criado para configurar o modo de funcionamento de um documento inteligente e todas as definições necessárias para este assentar em cima da framework da B-Simple. Tem um grande enfoque na definição do workflow dos módulos. Mais tarde, este módulo foi alterado para configurar qualquer módulo da framework.

Ao longo deste capítulo serão descritas as suas principais dimensões de configuração, correspondentes a vários tabuladores do ecrã.

### 4.1 Workflow

Para o conjunto de actividade de um documento inteligente é possível definir o seu workflow através de uma ferramenta gráfica. Na figura abaixo pode observar-se a configuração do exemplo que se vem seguindo.

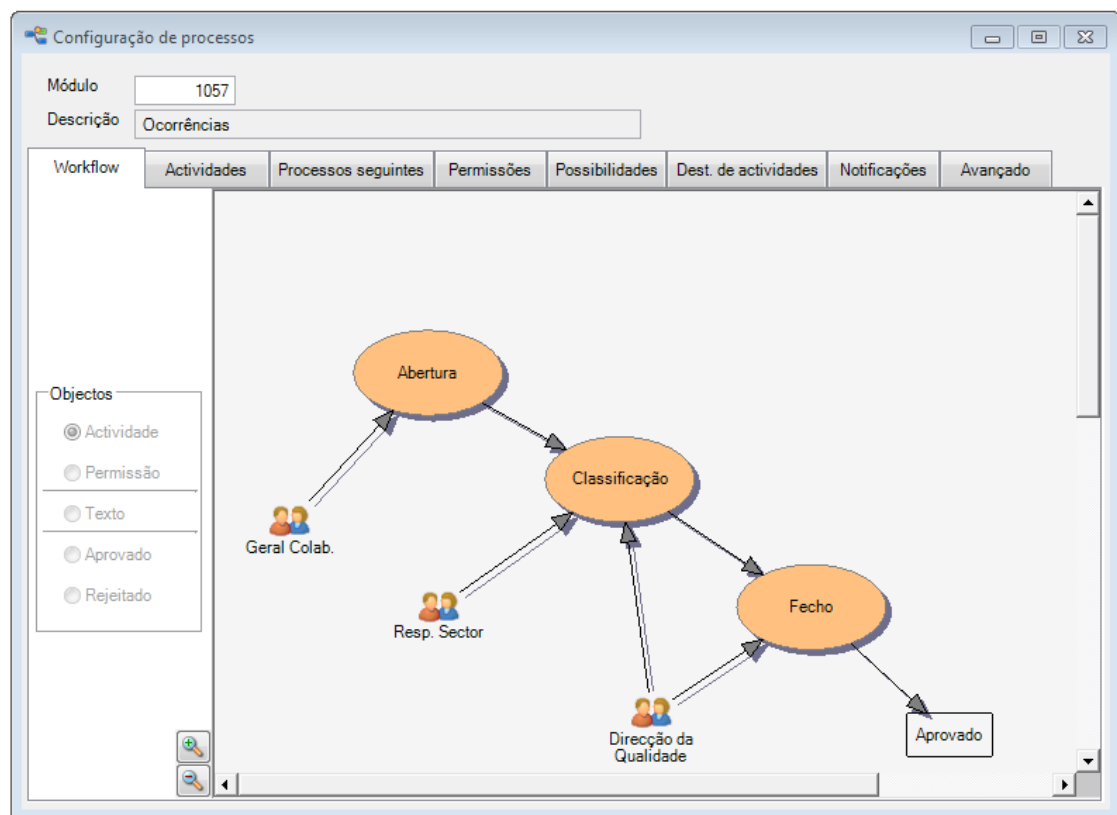


Figura 4.1 – Configuração do workflow

O ecrã é constituído pelos seguintes objectos:

- Actividades (objectos ovais)
- Permissões (pessoas)
- Texto
- Aprovado
- Rejeitado

A interacção consiste em ligar os objectos através do mecanismo de *drag&drop*. Esta ligação configura automaticamente a forma como o *workflow* acontece. Por exemplo:

Se uma seta liga a actividade Abertura à actividade Classificação, significa que após a conclusão da Abertura o sistema passa para a Classificação;

Se uma seta liga os colaboradores a uma actividade, significa que estes vão participar nessa actividade.

Em cada um dos objectos é no entanto possível definir opções mais avançadas com um click do rato em cima do respectivo objecto. É possível por exemplo:

- Definir quantas aprovações (colaboradores) são necessários para aprovar a actividade;
- Se existem actividades em paralelo que necessitem de ser aprovadas para passar ao passo seguinte;
- Duração máxima da actividade;
- Periodicidade das notificações aos intervenientes;
- Escrever um script para passagem à actividade seguinte;
- Possibilidade de definir quem vai executar a actividade no momento em que está a decorrer o processo.

## 4.2 Processos seguintes

A configuração permite definir uma ou várias ligações a processos seguintes. Esta ligação permite que a conclusão do processo actual automaticamente despolete outros processos, ou seja, outros documentos inteligentes. O ecrã abaixo mostra uma configuração em que após o fecho de uma ocorrência é iniciado o processo de uma Acção Correctiva e Preventiva (ACP), mostra ainda que o módulo é aberto automaticamente ao utilizador.



Figura 4.2 – Configuração dos processos seguintes

## 4.3 Possibilidades

Em cada actividade o utilizador tem de adoptar uma acção conclusiva. Por omissão o sistema permite quatro possibilidades:

- Em aberto: A tarefa fica por concluir;
- Aprovado: Dá a sua aprovação à actividade;
- Rejeitado: Rejeita o processo;
- Voltar atrás: Volta a actividade anterior.

No entanto é possível configurar outras alternativas recorrendo ao tabulador das Possibilidades (ver imagem abaixo).

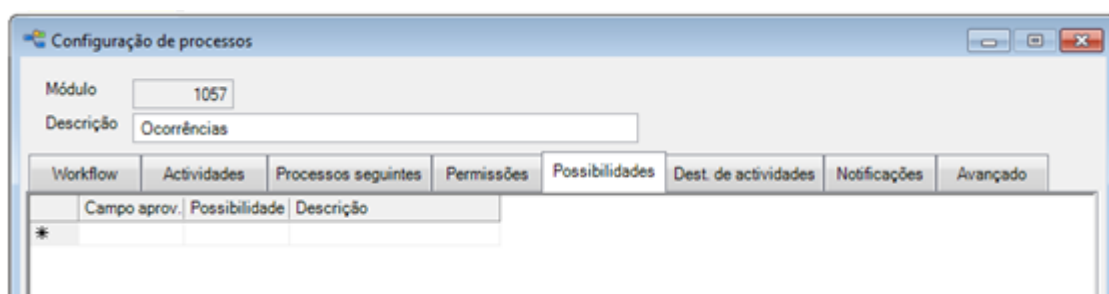


Figura 4.3 – Configuração das possibilidades por actividade

## 4.4 Notificações

O sistema envia automaticamente mensagens aos intervenientes que necessitam intervir no processo. No entanto, é também possível configurar mensagens (ver figura seguinte) para outros colaboradores que não estão a intervir directamente, mas necessitam de estar a par do processo.

Aqui é possível definir, por cada actividade, mensagens para diferentes destinatários (por colaborador ou responsabilidade) e a forma de entrega: SMS, Email ou B-Quality (próprio sistema).

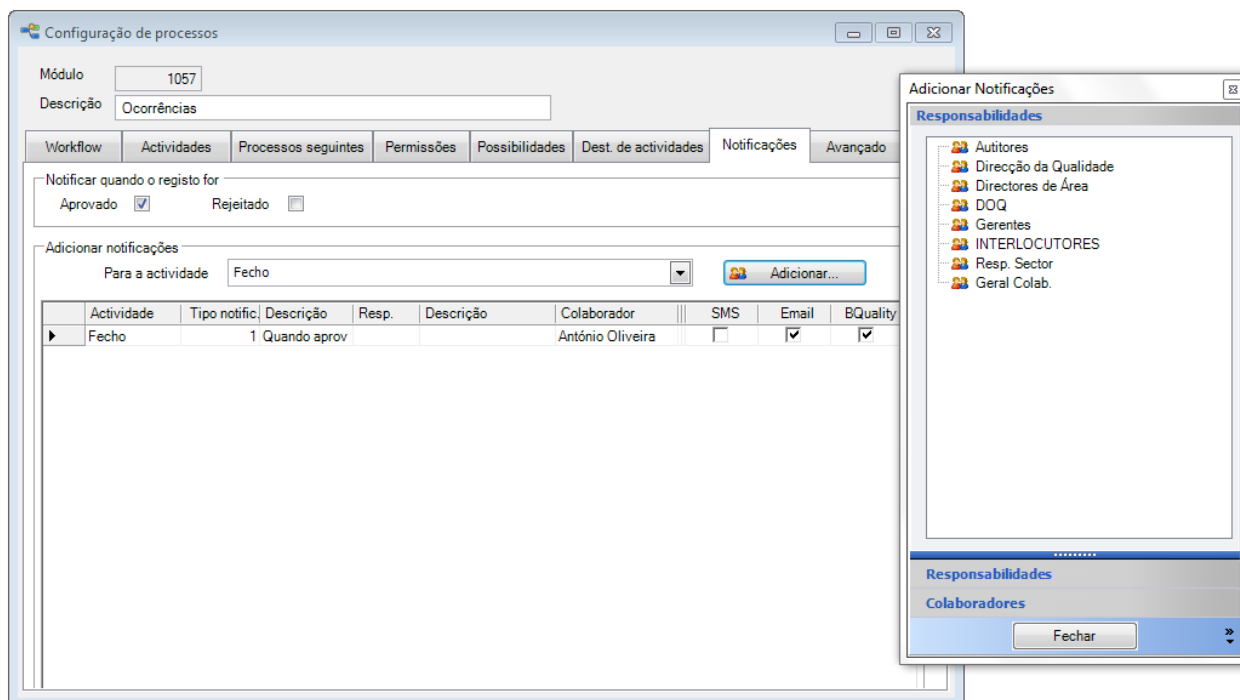


Figura 4.4 – Configuração das notificações

## 4.5 Avançado

O tabulador avançado permite efectuar configurações técnicas do documento inteligente, das quais se destaca:

- Nome do módulo: tem o nome da classe da dll;
- Se controla versões de informação;
- Criação do módulo no menu da aplicação;
- O URL do módulo no caso de web;
- O ficheiro de ajuda ao utilizador;
- Se tem uma instância única (singleton);
- As listagens que podem estar associadas ao módulo;
- Configuração das pesquisas que podem ser feitas pelo utilizador

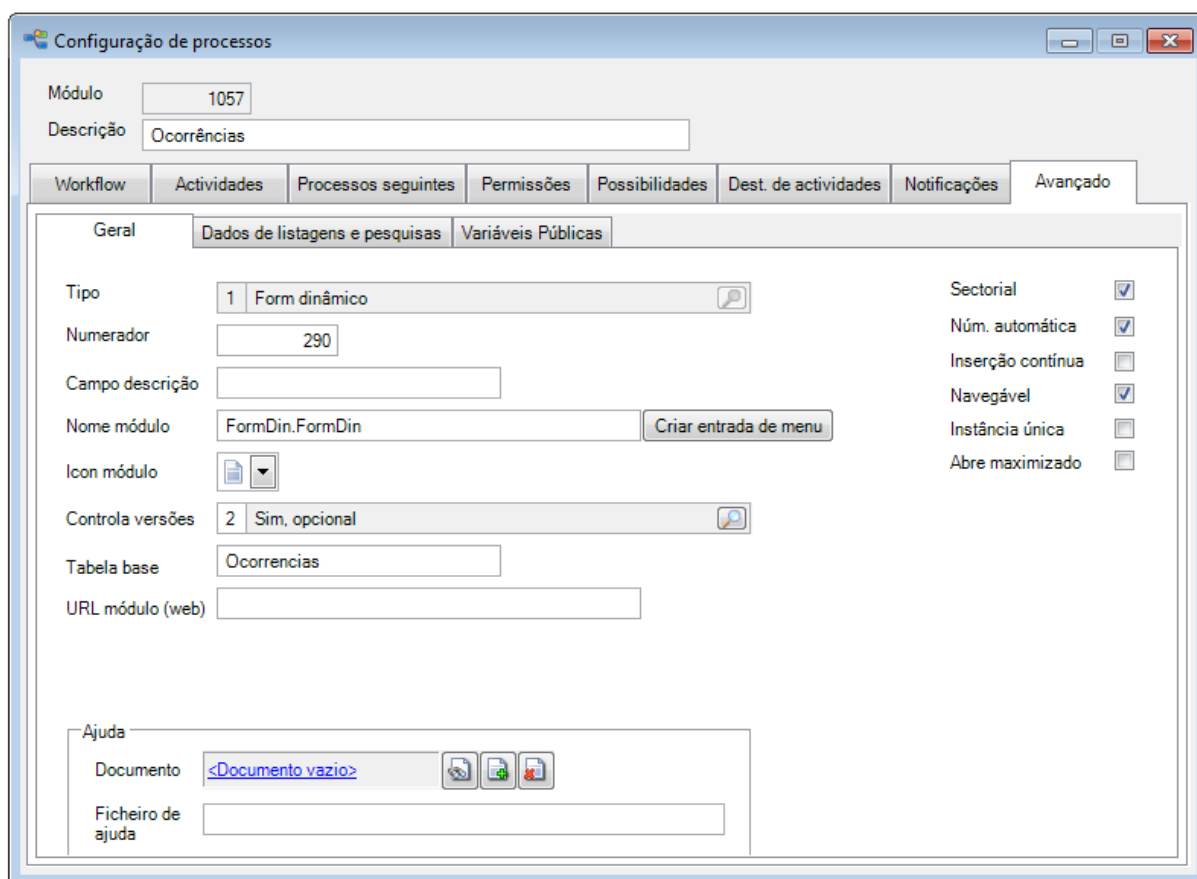


Figura 4.5 – Configuração das opções avançadas

## 5 CLASSES E COMPONENTES

Neste capítulo são apresentadas as principais classes usadas como suporte dos documentos inteligentes. No primeiro subcapítulo é apresentado o diagrama de classes e a descrição dos principais métodos e no segundo subcapítulo é descrita, em traços gerais, a interação entre as várias classes através de um diagrama de sequência. Por fim é apresentado o diagrama de componentes e a forma como se ligam com a framework da B-Simple.

### 5.1 Classes

Pretende-se com este subcapítulo apresentar, genericamente, a estrutura de classes envolvida nos documentos inteligentes.

É possível observar, no diagrama de classes seguinte, uma vista das classes principais envolvidas numa interação de um documento inteligente, onde apenas estão representados os principais métodos usados para explicar a interação. Refere-se o presente caso à criação de uma ocorrência.

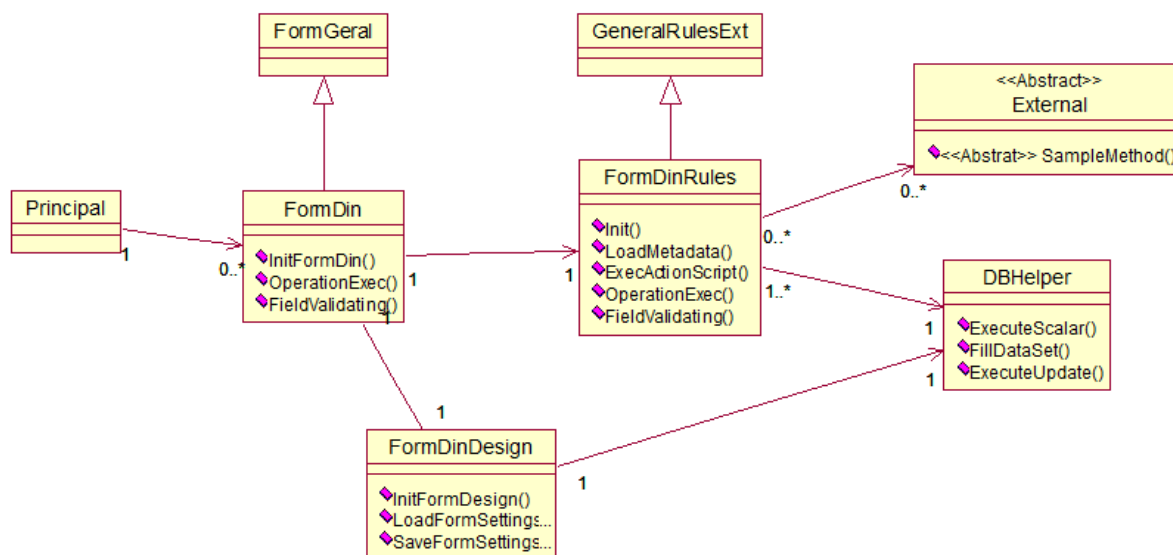


Figura 5.1 – Diagrama das classes principais

Nos próximos pontos são descritas as várias classes envolvidas e respectivos métodos.

### 5.1.1 Classe Principal

A classe Principal, como o próprio nome indica constitui a principal classe da framework B-Simple e representa o ponto de partida da aplicação: o executável. Esta classe funciona como o orquestrador das várias classes envolvidas na framework, como é o caso da invocação das funcionalidades de segurança, *logging* ou, conforme apresentado no diagrama anterior, a instanciação dos módulos.

### 5.1.2 Classe FormDin

A classe FormDin é a principal classe da camada de apresentação do documento inteligente, representando o *Form* de introdução da informação. É constituída por um conjunto de componentes visuais (*textbox*, *combobox*, *labels*, *images*, etc.), cujos métodos principais se encontram descritos na tabela seguinte.

Salienta-se, ainda, que esta classe obedece à estrutura da framework B-Simple através da herança da classe FormGeral, classe esta que é herdada por todos os módulos que trabalham em cima da framework.

Tabela 5.1 – Principais métodos do FormDin

Método	Descrição
InitFormDin	Inicializa o módulo e respectivos objectos associados, como, por exemplo, as regras de negócio.
OperationExec	Despoletada pelas operações típicas de interacção com o utilizador, como criar, alterar, apagar, gravar, cancelar e pesquisar. Normalmente invoca um método semelhante na camada das regras de negócio.
FieldValidating	Apanha os eventos de validação de cada campo do <i>Form</i> e invoca a sua validação na camada das regras de negócio.

### 5.1.3 Classe FormDinDesign

A classe FormDinDesign é a classe que dinamicamente desenha todo o *Form* de interacção com o utilizador. A sua criação obedece aos metadados definidos anteriormente pelo B-Designer.



Tabela 5.2 – Principais métodos do FormDinDesign

Método	Descrição
InitFormDesign	Inicializa o módulo e respectivos objectos associados, como por exemplo as regras de negócio.
LoadFormSettings	Efectua o carregamento dos metadados respeitantes ao desenho e aplica-os no form que é passado por parâmetro.
SaveFormSettings	Guarda os metadados na base de dados. Usado pelo B-Designer.

### 5.1.4 Classe FormDinRules

A classe FormDinRules é a classe que gere as regras de negócio dos documentos inteligentes. Efectua a validação de toda a informação introduzida pelo utilizador através das regras definidas no ModConfig e permite, simultaneamente, a sua extensibilidade através da invocação de SQL ou *assemblies* externas.

Salienta-se ainda que esta classe obedece à estrutura da framework B-Simple através da herança da classe GeneralRulesExt, sendo esta herdada por todas as classes de regras de negócio que trabalham em cima da framework.

Tabela 5.3 – Principais métodos do FormDinRules

Método	Descrição
Init	Inicializa o módulo, e respectivos objectos associados, e aplica as definições iniciais das regras de negócio.
LoadMetadata	Efectua o carregamento dos metadados respeitantes às regras de negócio. Estas definições existem no ModConfig.
ExecActionScript	Gere os <i>hotspots</i> dos documentos inteligentes. Permite a extensibilidade dos documentos inteligentes através da invocação externa de SQL ou métodos de <i>assemblies</i> externas (por reflection).
OperationExec	Valida se se pode entrar no modo (criar, alterar, etc.), invocado pelo <i>form</i> , nomeadamente as regras de negócio associadas.
FieldValidating	Valida as regras de negócio associados a cada input. Responde a cada pedido de validação do FormDin respeitante à introdução de dados.

### 5.1.5 Classe DBHelper

A classe DBHelper é a camada de acesso a dados. Aqui são definidas as operações típicas de manipulação da base de dados, tais como: Inserir, Alterar, Apagar e Consultar.

Esta classe contém também os métodos de mapeamento para carregamento das estruturas usadas na aplicação para suporte aos dados persistentes, que serão apresentados no próximo capítulo.

### 5.1.6 Classe External e Extensibilidade

A classe External pretende representar qualquer classe externa que seja instanciada pelos documentos inteligentes. Estas classes, e respectivos métodos, são configuradas no ModConfig e executadas por *reflection*. Esta é uma das possibilidades de extensibilidade dos documentos inteligentes, permitindo invocar situações que não foram previstas pela sua estrutura.

## 5.2 Diagrama de Sequência

Pretende-se neste subcapítulo descrever resumidamente a forma como as várias classes se relacionam por forma a manipular registos.

Como se pode observar no diagrama de sequência seguinte, a interacção é sempre iniciada pelo pedido da framework, através da classe Principal, para iniciar o módulo. Na sequência desse processo, a classe FormDin efectua as inicializações e, através do FormDinDesign, constrói e apresenta ao utilizador o *Form* para introdução de dados.

Em qualquer momento o utilizador pode desencadear uma operação (novo, alterar, anular, gravar, etc.) que, através do método OperationExec, prepara as classes para a operação, efectuando ainda a validação da operação. Por exemplo, ao não admitir a alteração de um registo.

Outra acção que é invocada sempre que o utilizador faz uma introdução de dados é a validação da informação. Esta é feita pelos métodos de *validating*.

A interacção termina quando existe uma operação de Gravar ou Cancelar que é também feita através do método OperationExec.

Destaca-se ainda a possibilidade de extensibilidade, já atrás referida neste capítulo, que é iniciada pelo método `ExecActionScript`, permitindo a invocação de um comando SQL ou de um método de uma *assemblie* externa.

Repare-se que esta extensibilidade pode ser despoletada em vários pontos da interacção.

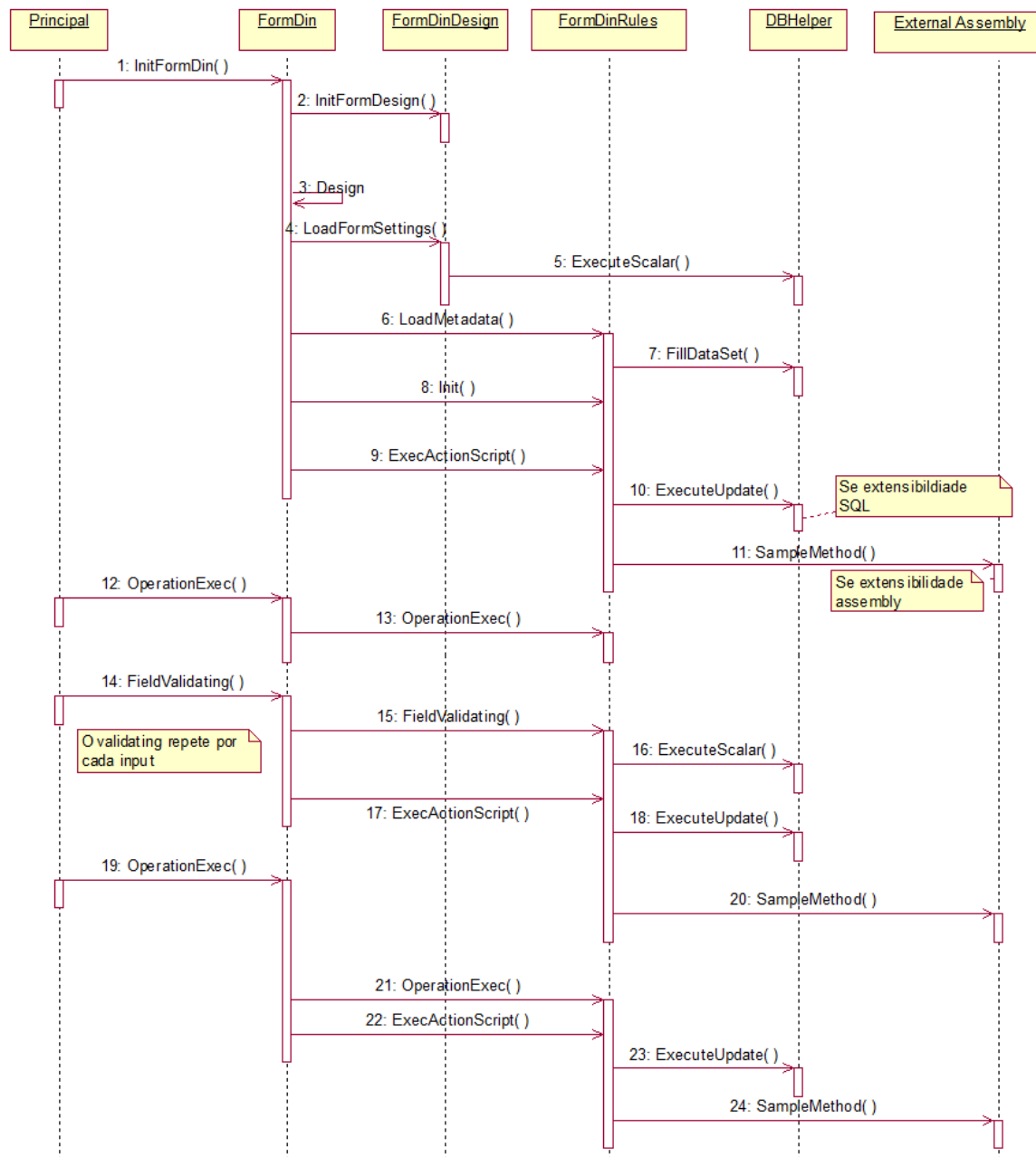


Figura 5.2 – Diagrama de sequência de criação de um registo

### 5.3 Diagrama de Componentes

Pretende-se neste subcapítulo descrever resumidamente os vários componentes que resultaram do desenvolvimento dos documentos inteligentes e as suas dependências para com a framework.

No diagrama abaixo é possível observar os vários componentes envolvidos. A amarelo estão representados alguns dos componentes da framework e a laranja os novos componentes desenvolvidos para os documentos inteligentes.

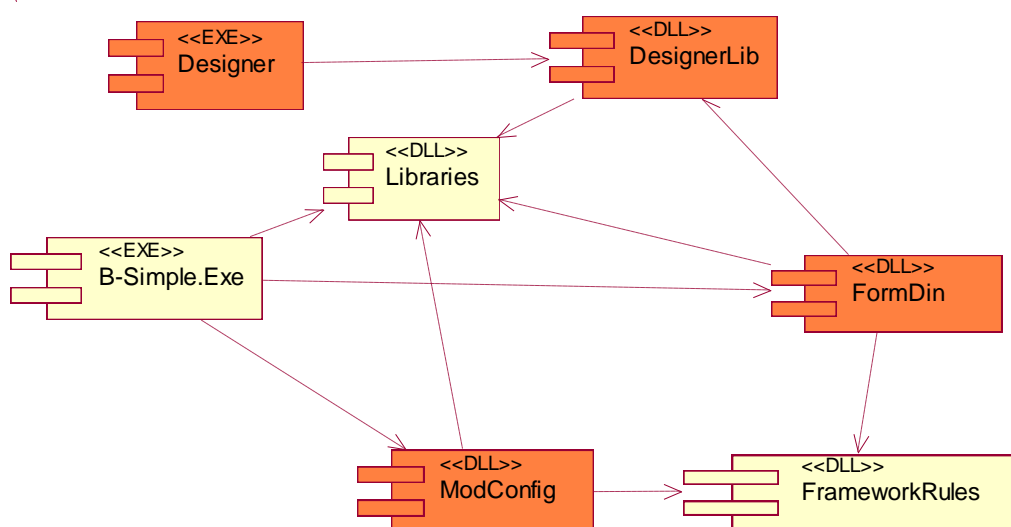


Figura 5.3 – Diagrama de sequência de criação de um registo

Tabela 5.4 – Componentes envolvidos

Método	Descrição
Designer	É o executável que representa o B-Designer. Tem uma ligação com o DesignerLib para a leitura e gravação dos metadados.
DesignerLib	Contém, entre outras, a classe DesignerLib que efectua o desenho dinâmico do Form.
Libraries	Contém as funcionalidades/utilitários comuns à maioria dos módulos da framework.
B-Simple.Exe	É o executável da framework. Contém entre outras a classe Principal.
FormDin	Contém as principais classes usadas para a execução do documento inteligente, nomeadamente a FormDin e a

FormDinRules.	
ModConfig	Contém as classes responsáveis pela configuração do documento inteligente, nomeadamente a ModConfigForm e a ModConfigRules.
FrameworkRules	Contém as classes comuns a vários módulos e as classes que representam as funcionalidades transversais da framework. Uma das classes aqui presente é a GeneralRulesExt e o DBHelper.

## 6 MODELO DE DADOS

Neste capítulo são apresentadas as principais tabelas usadas para suporte aos documentos inteligentes. No primeiro subcapítulo é apresentado o modelo de dados respeitante à configuração dos documentos inteligentes e no segundo subcapítulo são apresentadas as tabelas abstractas que cada documento inteligente específico tem.

### 6.1 Tabelas de configuração

Para configuração de todos os documentos inteligentes, como por exemplo o workflow, os intervenientes e os alertas, foi usado um conjunto de tabelas que são apresentadas no diagrama de classes persistentes seguinte.

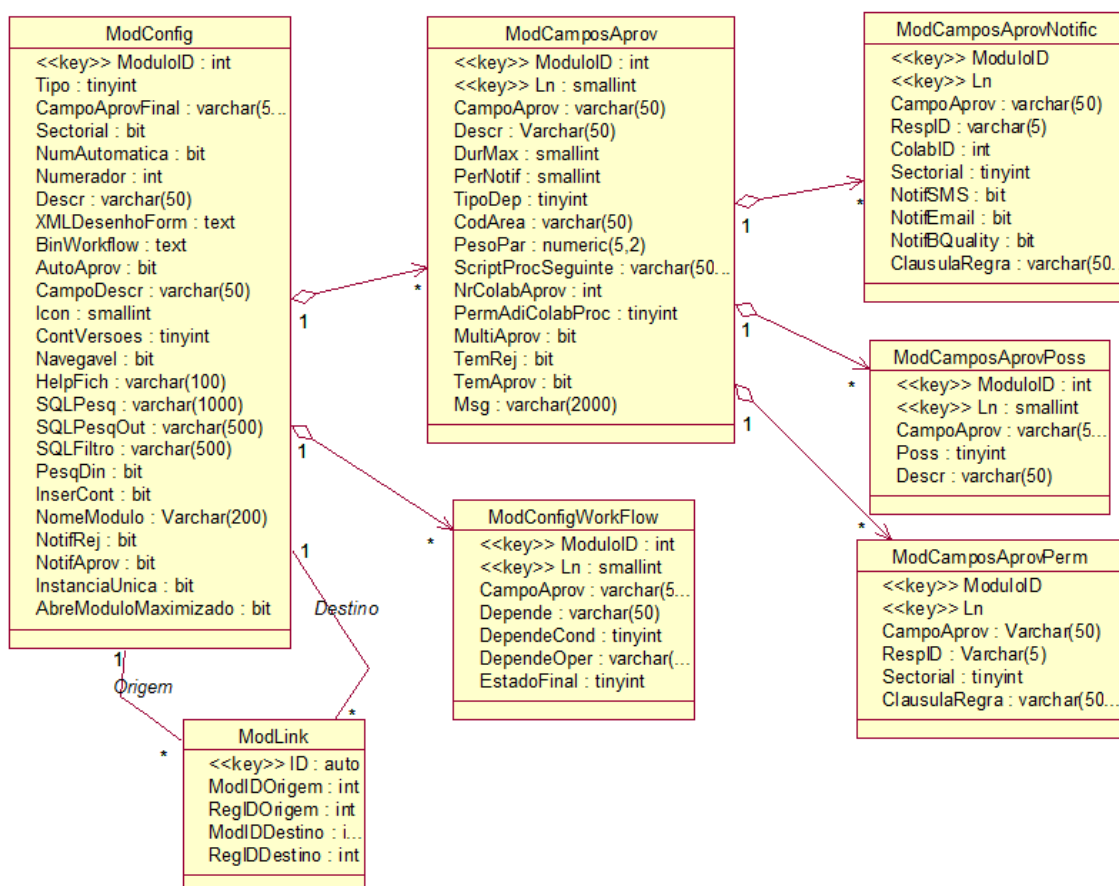


Figura 6.1 – Diagrama das classes persistentes de configuração

Nas tabelas seguintes é possível observar o significado de cada um dos campos apresentados no diagrama.

### 6.1.1 Tabela ModConfig

A tabela ModConfig é a tabela master das configurações. Aqui ficam registados os principais atributos que ocorrem apenas uma vez por módulo.

Tabela 6.1 - Campos do ModConfig

Campo	Descrição
ModuloID	Identifica unicamente o módulo.
Tipo	Tipo do módulo: 0 - Normal (default) 1 – Documento inteligente 2 - Listagem 3 - Formulário para workflow documental 4 - Executável externo - executa uma shell
CampoAprovFinal	Indica qual o nome do campo de aprovação final, não obrigatório (pois podem existir módulos que nem têm processos de aprovação).
Sectorial	Indica se o módulo tem conceitos sectoriais. Se sim, o módulo tem obrigatoriamente de ter um campo SectorID  Aquando da definição de permissões e notificações é importante perguntar ao utilizador se é ou não sectorial.
NumAutomatica	Se o módulo tem numeração automática.
Numerador	Qual o numerador quando a numeração é automática.
Descr	Descrição do módulo.
XMLDesenhoForm	Campo que guarda a informação necessária para representar o desenho de um formulário e todos os seus componentes.  Este campo é actualizado quando é gravado um formulário através do B-Designer.
BinWorkflow	Guarda os objectos e a sua disposição em termos de workflow.
AutoAprov	Se aprova automaticamente o registo. Útil quando não existe processo de aprovação.
CampoDescr	Nome do campo Descrição.  Ex. Nos sectores é "Descr" e nos colaboradores é "Nome".

Icon	Nr do icon que fica associado ao módulo. Tem de existir na lista da framework.
ContVersoes	Se o módulo tem controlo de versões: 0 - Não controla 1 - Controla sem dar hipótese de alterar 2 - Pergunta ao utilizador se gera ou não uma nova versão
Navegavel	Se o módulo é navegável, ou seja, se nos navegadores é dada a possibilidade de adicionar o módulo.
HelpFich	Ficheiro de help em disco. Normalmente tipo .chm. Este ficheiro deve ser aberto com um ID de help que vem do módulo.
SQLPesq	SQL usado para fazer o match dos valores com as variáveis e ser acrescentado à cláusula where. Apenas faz sentido quando PesqDin=1, ou seja quando o módulo usa pesquisa dinâmica.
SQLPesqOut	Define os campo que devem surgir na grelha de pesquisa. SELECT <campos> FROM <Tabela> Ex. Select Data, Descr, (select x from art) Art From Art Se não preenchido aparece: Código + Descrição Apenas faz sentido quando PesqDin=1
SQLFiltro	Filtros para acesso à informação dos módulos. Aqui pode ser definida uma cláusula que se estiver activa nos utilizadores, obriga a ser executada em todas as pesquisas. Estão disponíveis as variáveis do DatasetEmpresa e o colaborador.
PesqDin	Se a pesquisa dos módulos é dinâmica. Se não assume os especificado no módulo.
InserCont	Se tem ou não inserção continua de dados.
NomeModulo	Nome módulo.
NotifRej	Se notifica os colaboradores em caso de rejeição (os que já tinham intervindo).
NotifAprov	Se notifica os colaboradores quando o processo chega ao fim como aprovado (os que já tinham intervindo).
InstanciaUnica	Controla se um determinado módulo, sempre que é invocado, abre sempre ou é um singleton.



AbreModuloMaximizado	Se igual a 1, condiciona o formulário a abrir em modo especial.  Este modo especial define uma barra do <i>form</i> com estilo diferente, bem como obriga o formulário a abrir maximizado não sendo possível visualizá-lo noutro estado.
----------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 6.1.2 Tabela ModLink

A tabela ModLink permite guardar a ligação entre documentos inteligentes.

Tabela 6.2 - Campos do ModLink

Campo	Descrição
ID	Identifica único sequencial (automático).
ModIDOrigem	Módulo do processo origem.
RegIDOrigem	ID do Registo origem.
ModIDDestino	Módulo destino.
RegIDDestino	ID do Registo destino

### 6.1.3 Tabela ModConfigWorkFlow

A tabela ModConfigWorkFlow guarda as definições das actividades do workflow.

Tabela 6.3 - Campos do ModConfigWorkFlow

Campo	Descrição
ModuloID	A que módulo se refere.
Ln	Sequencial dentro do módulo.
CampoAprov	Qual o campo da actividade que define o estado da actividade.
Depende	Regista a dependência de outra actividade. Os que não têm dependência, não ficam aqui configurados.

	Depende do campo DependCond.
DependeCond	Define o workflow de forma condicional. É obrigatório estar preenchido, default = 1 (aprovado).
DependeOper	Operador a testar: =, <>, >, <, >= ou <=
EstadoFinal	Este campo apenas é usado na passagem para o estado final. Tem 2 valores possíveis: 1 - Aprovado 2 - Rejeitado

#### 6.1.4 Tabela ModCamposAprov

A tabela ModCamposAprov regista os campos de aprovação que tem o módulo. Nos documentos inteligentes é possível registá-los no B-Designer e/ou no B-Config.

Tabela 6.4 - Campos do ModCamposAprov

Campo	Descrição
ModuloID	A que módulo se refere.
Ln	Sequencial dentro do módulo.
CampoAprov	Qual o campo da aprovação da actividade.
Descr	Descrição dada ao campo de aprovação, para poder ser visualizada no workflow, notificações, etc.
DurMax	Duração máxima em horas que deverá ter para ser aprovado (ou rejeitado).  Default em Empresa.DurMax  Pode ser alterada mais tarde num programa de configuração.  Default=24
PerNotif	Periodicidade da notificação dos processos pendentes em horas.

TipoDep	<p>Tipo de dependência:</p> <p>1 - Dependência tipo E (ou seja todos os campos do workflow têm de estar a 1 para poder avançar) (default)</p> <p>2 - Dependência tipo OU (ou seja basta um campo do workflow estar a 1 para poder avançar)</p>
CodArea	<p>Área do formulário. Pode ser apenas uma edit. Ex. GroupBox1</p> <p>Apenas para controlo da interface. Relaciona as áreas de interface com os campos de aprovação.</p> <p>Para uma área estar activa é necessário verificar no Workflow se todas as dependências estão a 1.</p>
PesoPar	<p>Peso em actividades paralelas. Default 100%.</p> <p>Quando existem várias actividades em paralelo apenas avança quando a soma de todas for maior ou igual a 100%.</p>
ScriptProcSeguinte	<p>Este atributo surge com o intuito de definir qual o processo seguinte que fica disponível. É despoletado sempre que o estado muda para Aprovado.</p>
NrColabAprov	<p>Nr de colaboradores necessários para aprovar o processo. Usado na aprovação múltipla. Se zero - todos.</p>
PermAdiColabProc	<p>Se o colaborador pode adicionar mais colaboradores ao processo actual:</p> <p>0 - Não</p> <p>1 – Pode</p>
MultiAprov	<p>Se a actividade é multi-aprovação.</p> <p>Significa que este passo do processo, só estará definitivamente aprovado se o NrColabAprov for atingido (Se NrColabAprov=0 então todos tem de aprovar).</p> <p>Uma das obrigаторiedades desta classificação é que exista uma tabela denominada &lt;TabelaBase&gt;MultiAprov.</p>

	Nesta tabela serão guardados os estados de aprovação parciais para cada colaborador/actividade.
TemRej	Se o campo tem a opção de rejeitar. Isto porque há casos em que apenas faz sentido aprovar (verificar).
TemAprov	Se o campo tem a opção de aprovação.
Msg	Mensagem que irá fazer parte da notificação gerada a partir do processo pendente.  Caso o campo esteja preenchido vai ser usado como mensagem da notificação.

### 6.1.5 Tabela ModCamposAprovNotific

A tabela ModCamposAprovNotific permite definir quem notificar em cada actividade e a forma de o fazer.

Tabela 6.5 - Campos do ModCamposAprovNotific

Campo	Descrição
ModuloID	A que módulo se refere.
Ln	Sequencial dentro do módulo.
CampoAprov	Qual o campo da actividade que define o estado da actividade.
RespID	Que colaboradores (com estas responsabilidades) vão ser notificados.
ColabID	Qual é o colaborador que vai ser notificado. Se o campo RespID<>" então este fica a zero.
Sectorial	Se apenas notifica quem está ligado ao sector:  0 - Não – Normal - notifica todos os definidos pelas responsabilidades  1 - Apenas quem está ligado ao sector

2 - Apenas quem é responsável pelo sector em causa	
NotifSMS	Se a notificação é feita por SMS.
NotifEmail	Se a notificação é feita por Email.
NotifBQuality	Se a notificação é feita pela aplicação.
ClausulaRegra	Condiciona notificações manuais associadas a um processo

### 6.1.6 Tabela ModCamposAprovPoss

A tabela ModCamposAprovPoss permite definir as possibilidades de aprovação de uma actividade para além das *standards*.

Tabela 6.6 - Campos do ModCamposAprovPoss

Campo	Descrição
ModuloID	A que módulo se refere.
Ln	Sequencial dentro do módulo.
CampoAprov	Qual o campo da actividade que define o estado da actividade.
Poss	ID da possibilidade.
Descr	Descrição da possibilidade.

### 6.1.7 Tabela ModCamposAprovPerm

A tabela ModCamposAprovPerm permite definir as permissões por actividade. As permissões são definidas por responsabilidade.

Tabela 6.7 - Campos do ModCamposAprovPerm

Campo	Descrição
ModuloID	A que módulo se refere.
Ln	Sequencial dentro do módulo.
CampoAprov	Qual o campo da actividade que define o estado da actividade.
RespID	Que responsabilidades podem actuar nesta actividade. Se o campo não estiver preenchido, significa todas.
Sectorial	Se apenas se aplica a quem está ligado ao sector: 0 - Não – Normal - Todos os definidos pelas responsabilidades 1 - Apenas quem está ligado ao sector 2 - Apenas quem é responsável pelo sector em causa
ClausulaRegra	Serve para definir o acesso aos campos de aprovação por colaborador mas mediante uma condição prévia. A aplicação faz <i>parsing</i> da cláusula substituindo @<Campo> pelo valor do campo para o registo actual. Se o <i>script</i> devolver linhas considera-se a expressão verdadeira e portanto é dado o acesso para aprovar o processo.

## 6.2 Tabelas por documento inteligente

Aquando da criação de um documento inteligente é criado automaticamente um conjunto de tabelas e campos definidos como obrigatórios para o funcionamento do processo. Estas tabelas podem ser observadas no diagrama seguinte, tendo em consideração que a nomenclatura <Modulo> é substituída pelo nome real do módulo.

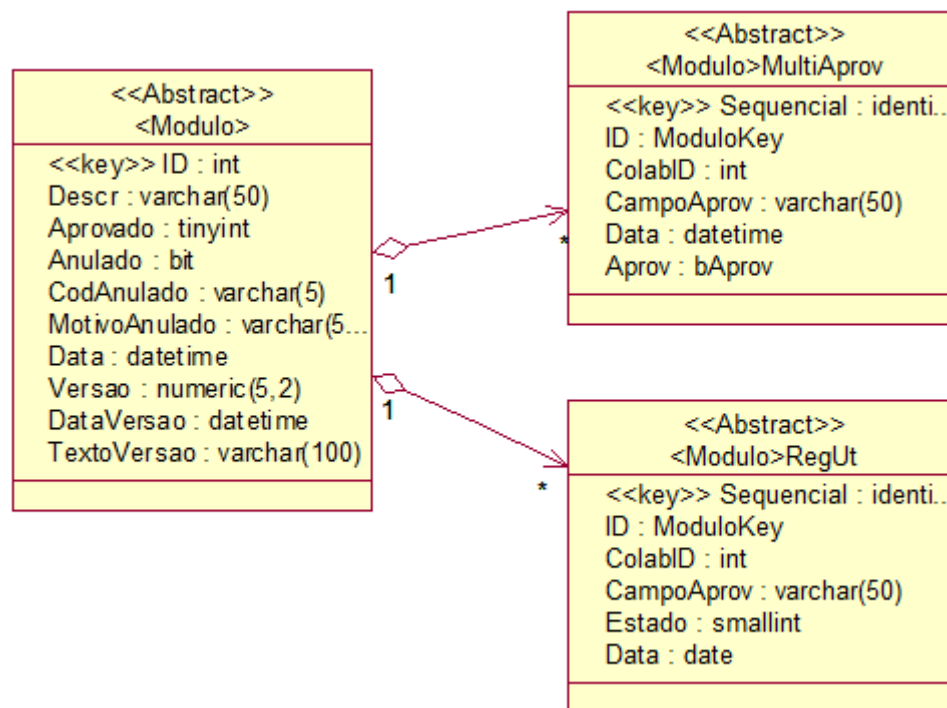


Figura 6.2 – Diagrama das classes persistentes modelo

### 6.2.1 Tabela <Modulo>

A tabela <Modulo> contém os campos obrigatórios que devem existir na tabela master do processo.

Tabela 6.8 - Campos do &lt;Modulo&gt;

Campo	Descrição
ID	Identificador único do módulo.
Descr	Descrição do registo.
Aprovado	<p>Todos os módulos têm de ter um campo Aprovado com pelo menos os seguintes estados:</p> <p>0 - Em aberto</p> <p>1 - Aprovado (actualizado quando todos os campos CampoAprov que não tenham processo seguinte estejam a 1)</p> <p>2 - Rejeitado (actualizado quando pelo menos um dos CampoAprov = 2)</p> <p>3 - Anulado (Quando o registo é anulado)</p> <p>7 - Proc. Actual - Significa que está a decorrer</p>
Anulado	<p>Se o registo foi anulado.</p> <p>Se anular, todos os colaboradores que estiveram em processos anteriores devem receber uma notificação</p> <p>Neste caso o campo Aprovado fica também com o estado 3 - anulado.</p>
CodAnulado	Classificação da anulação. Tem de existir na tabela CodAnul.
MotivoAnulado	Descrição da anulação.
Data	Data do registo
Versao	<p>Versão do registo.</p> <p>Usado quando ModConfig.ContVersoes&gt;0.</p>
DataVersao	Data início de versão.
TextoVersao	Texto que o utilizador pode inserir aquando da criação de uma nova versão do registo.
DataCriacao	Data de criação do registo.
UtCriacao	Utilizador que criou o registo.
DataAlt	Data de última alteração do registo.
UtAlt	Último utilizador a alterar o registo.



### 6.2.2 Tabela <Modulo>MultiAprov

A tabela <Modulo>MultiAprov é a tabela modelo para módulos que incluem actividades de multi-aprovação.

Tabela 6.9 - Campos do <Modulo>MultiAprov

Campo	Descrição
ID	ID do registo da tabela base.
ColabID	Colaborador que actua nesta actividade.
CampoAprov	Actividade. Num determinado momento do processo poderão existir várias actividades referenciadas ao mesmo tempo nesta tabela. Isto acontece no caso em que são definidos destinatários de actividades para frente durante o processo.
Data	Data de aprovação/rejeição deste colaborador. Inclui a hora.
Aprov	Estado da aprovação do colaborador: 0 - Em aberto 1- Aprovado 2- Rejeitado

### 6.2.3 Tabela <Modulo>RegUt

A tabela <Modulo>RegUt é a tabela modelo para o registo de todas as alterações por utilizador e por actividade.

Tabela 6.10 - Campos do <Modulo>RegUt

Campo	Descrição
ID	ID do registo da tabela base.
ColabID	Colaborador que actuou nesta actividade.
CampoAprov	Actividade.

Sequencial	Identificador único do registo (sequencial).
Estado	Estado: 1 - Aprovado 2 - Rejeitado
Data	Data em que aconteceu a mudança de estado (inclui hora).

## 7 CONCLUSÕES

Este trabalho pretende dar a conhecer uma visão geral do projecto dos documentos inteligentes e o seu enquadramento na framework de desenvolvimento da B-Simple. Ficaram por referir vários detalhes que, apesar da sua importância, tornariam o trabalho extenso e implicariam uma maior complexidade desta apresentação. Espera-se que, com esta apresentação, se tenha conseguido demonstrar o conceito por detrás deste tema.

### 7.1 Ferramentas e metadados

As ferramentas desenvolvidas vieram comprovar a sua grande eficácia. Com o B-Designer e o B-Config torna-se fácil desenhar um formulário e configurar um *workflow* e todas as suas propriedades com relativa facilidade e rapidez.

A interacção criada com o B-Designer e o configurador de workflow através de opções de *drag&drop* disponibilizam ao utilizador/configurador um ambiente simples para a criação de processos, sem recorrer a codificação.

A opção dos metadados gerados a partir do B-Designer ficarem separados da parte tecnológica facilitou a evolução do projecto e migrações entre diferentes bases de dados. A comprovar esta afirmação, existe actualmente a opção dos mesmos metadados servirem para criar um documento inteligente em Windows forms ou em ASP.

### 7.2 Mercado

Uma das preocupações iniciais do projecto era responder a diferentes problemas em diferentes tipos de organizações. Podemos comprovar o sucesso da solução através da diversidade de organizações onde conseguiu ser implementado. Actualmente existem implementações nos seguintes sectores:

- Alimentar
- Bancário
- Certificação e auditoria
- Confecção
- Consultoria

- Desenvolvimento de software
- Logística
- Saúde
- Transportes
- Turismo

### 7.3 Considerações finais

Pode-se concluir que a solução desenvolvida deu resposta ao problema inicial, pois é possível criar rapidamente um processo na organização à medida das suas necessidades e por outro lado garantir a independência dos implementadores com a software house.

Esta solução permite actualmente aos conhecedores do negócio implementar facilmente processos na área da qualidade sem terem a necessidade de grandes conhecimentos tecnológicos. Por outro lado, a ferramenta é também muito usada para desenvolvimento rápido(RAD) de módulos fora deste domínio.

O trabalho consegue de uma forma relativamente simples integrar muitos conceitos num único sistema de informação, sejam eles a engenharia de domínio, uma ferramenta RAD, o workflow, a abstracção ou desenvolvimento por camadas, podendo, assim afirmar-se que este vale pelo seu todo.